

Robust Localization from Incomplete Local Information

Amin Karbasi, *Student Member, IEEE*, Sewoong Oh, *Member, IEEE*

Abstract

We consider the problem of localizing wireless devices in an ad-hoc network embedded in a d -dimensional Euclidean space. Obtaining a good estimation of where wireless devices are located is crucial in wireless network applications including environment monitoring, geographic routing and topology control. When the positions of the devices are unknown and only local distance information is given, we need to infer the positions from these local distance measurements. This problem is particularly challenging when we only have access to measurements that have limited accuracy and are incomplete. We consider the extreme case of this limitation on the available information, namely only the connectivity information is available, i.e., we only know whether a pair of nodes is within a fixed detection range of each other or not, and no information is known about how far apart they are. Further, to account for detection failures, we assume that even if a pair of devices is within the detection range, it fails to detect the presence of one another with some probability and this probability of failure depends on how far apart those devices are. Given this limited information, we investigate the performance of a centralized positioning algorithm MDS-MAP introduced by Shang et al. [SRZF03], and a distributed positioning algorithm HOP-TERRAIN introduced by Savarese et al. [SLR02]. In particular, for a network consisting of n devices positioned randomly, we provide a bound on the resulting error for both algorithms. We show that the error is bounded, decreasing at a rate that is proportional to R_{Critical}/R , where R_{Critical} is the critical detection range when the resulting random network starts to be connected, and R is the detection range of each device.

Amin Karbasi is with the School of Computer and Communication Sciences, Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland (email: amin.karbasi@epfl.ch). Sewoong Oh is with the department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology (MIT), Cambridge, MA 02139 (email: swoh@mit.edu). A preliminary summary of this work appeared in [OKM10], [KO10].

I. INTRODUCTION

In this paper, we address the problem of positoining (also referred to as sensor localization) when only a set of incomplete pairwise distances is provided. Location estimation of individual nodes is a requirement of many wireless sensor networks such as environment monitoring, geographic routing and topology control, to name only a few (for a thorough list of applications we refer the interested readers to [JH01], [Xu02]). In environment monitoring for instance, the measurement data by the wireless sensor network is essentially meaningless without knowing from where the data is collected.

One way to acquire the positions is to equip all the sensors with a global positioning system (GPS). The use of GPS not only adds considerable cost to the system, but more importantly, it does not work in indoor environments or when the received GPS signal is jammed (see [CHH02] and the references therein for more information on this issue). As an alternative, we seek an algorithm that can derive positions of sensors based on local/basic information such as proximity (which nodes are within communication range of each other) or local distances (pairwise distances between neighbouring sensors).

Two common techniques for obtaining the local distance and connectivity information are Received Signal Strength Indicator (RSSI) and Time Difference of Arrival (TDoA). RSSI is a measurement of the ratio of the power present in a received radio signal and a reference power. Signal power at the receiving end is inversely proportional to the square of the distance between the receiver and the transmitter. Hence, RSSI has the potential to be used to estimate the distance and it is common to assume the use of RSSI in distance measurements. However, experimental results indicate that the accuracy of RSSI is limited [PCB00]. TDoA technique uses the time difference between the receipt of two different signals with different velocities, for instance ultrasound and radio frequency signals [SHS01]. The time difference is proportional to the distance between the receiver and the transmitter; and given the velocity of the signals, the distance can be estimated from the time difference. These techniques can be used, independently or together, for distance estimation. In an alternative approach, Angle of Arrival (AoA) can also be used to infer the positions of sensors [NN01]. Once a node has the angle of arrival information from three other nodes with known positions, we can perform triangulation to locate the wireless node. To measure the angle of arrival, an antenna array is required at each wireless node.

Given a set of measurements, the problem of localization is solvable, meaning that it has a unique set of coordinates satisfying the given local information, only if there are enough constraints. The simplest of such algorithms, i.e., multi dimensional scaling (MDS) [BG05], assumes that all pairwise distances are known. Intuitively, it is clear that with $O(n^2)$ pairwise distances we should be able to determine

$O(n)$ coordinates. However, in almost all practical scenarios such information is unavailable for two major reasons. First, sensors are typically highly resource-constrained (e.g., power) and have limited communication range. Thus, far away sensors cannot communicate and obtain their pairwise distances. Second, due to noise and interference among sensors, there is always the possibility of non-detection or completely incoherent measurements.

Many algorithms have been proposed to resolve these issues by using heuristic approximations to the missing distances, and their success has mostly been measured experimentally. Regarding the mechanisms deployed for estimating sensor locations, one can divide the localization algorithms into two categories: range-based and range-free. In the range-based protocols the absolute point-to-point distance estimates are used for inferring the locations, whereas in the range-free protocols no assumptions about the availability of such information are made and only the connectivity information is provided. As a result, range-free algorithms are more effective in terms of stability and cost, hence more favourable to be deployed in practical settings.

The theoretical guarantees associated with the performance of the existing methods are, however, of the same interest and complementary in nature. Such analytical bounds on the performance of localization algorithms can provide answers to practical questions: for example, "How large should the radio range be in order to get the reconstruction error within a threshold?" With this motivation in mind, our work takes a step forward in this direction.

We first focus on providing a bound on the performance of a popular localization algorithm MDS-MAP [SRZF03] when applied to sensor localization from only connectivity information. We should stress here that pairwise distances are invariant under rigid transformations (rotation, translation and reflection). Hence, given connectivity information, we can only hope to determine the *configuration* or the relative map of the sensors. In other words, localization is possible only up to rigid transformations. With this point in mind, we prove that using MDS-MAP, we are able to localize sensors up to a bounded error in a connected network where most of pairwise distances are missing and only local connectivity information is given.

More precisely, assume that the network consists of n sensors positioned randomly in a d -dimensional unit cube with the radio range $R = o(1)$ and detection probability p . Let the $n \times d$ matrices X and \hat{X} denote the true sensor positions and their estimates by MDS-MAP, respectively. Define $L = \mathbb{I}_{n \times n} - (1/n)\mathbb{1}_n\mathbb{1}_n^T$ where $\mathbb{I}_{n \times n}$ is the identity matrix and $\mathbb{1}_n$ is the all ones vector. It is not difficult to show that LXX^TL satisfies nice properties, specifically, it is invariant under rigid transformations and if $LXX^TL = L\hat{X}\hat{X}^TL$, then X and \hat{X} are equal up to rigid transformations. Therefore, we can naturally

define the distance between X and \hat{X} as follows:

$$d_{\text{inv}}(X, \hat{X}) = \frac{1}{n} \|LX X^T L - L\hat{X} \hat{X}^T L\|_F,$$

where $\|\cdot\|_F$ denote the Frobenius norm. Our first result establishes a bound on the error of MDS-MAP in terms of d_{inv} , specifically,

$$d_{\text{inv}}(X, \hat{X}) \leq \frac{R_{\text{MDS}}}{R} + o(1),$$

where $R_{\text{MDS}} = C_d(\ln(n)/n)^{1/d}$ for some constant C_d that only depends on the dimension d .

One consequence of the ad-hoc nature of the underlying networks is the lack of a central infrastructure. This fact prevents the use of common centralized positioning algorithms such as MDS-MAP. In particular, centralized algorithms suffer from the scalability problem, and generally it is not feasible for them to be implemented in large scale sensor networks. Other disadvantages of centralized algorithms, as compared to distributed algorithms, are their requirements for higher computational complexity and lower reliability; these drawbacks are due to accumulated information inaccuracies caused by multi-hop transmission over a wireless network [MFA07].

We then investigate an important question about whether similar performance guarantees can be obtained in a distributed setting where each sensor tries to estimate its own *global* position. As mentioned above, this task cannot be accomplished unless some additional information, rather than local measurements, is provided. It is well known that in a d -dimensional Euclidean space, we need to know the global positions of at least $d + 1$ sensors, referred to as *anchors*, in order to uniquely determine the global positions of the remaining sensors [NN01].

For the decentralized scenario, we turn our attention to analysing the performance of a popular localization algorithm called HOP-TERRAIN algorithm [SLR02]. This algorithm can be seen as a distributed version of the MDS-MAP. Similar to MDS-MAP, we prove that by using HOP-TERRAIN, we are able to localize sensors up to a bounded error in a connected network where most of the pairwise distances are unknown and only local connectivity information is given.

More formally, assume that the network consists of n sensors positioned randomly in a d -dimensional unit cube and $d + 1$ anchors in general positions. Moreover, we let the radio range $R = o(1)$ and denote the detection probability by p . We show that when only connectivity information is available, for every unknown node i , the Euclidean distance between the estimate \hat{x}_i and the correct position x_i is bounded by

$$\|x_i - \hat{x}_i\| \leq \frac{R_{\text{HOP}}}{R} + o(1),$$

where $R_{\text{HOP}} = C'_d(\log n/n)^{\frac{1}{d}}$ for some constant C'_d that only depends on d .

TABLE I
DISTRIBUTED LOCALIZATION ALGORITHM CLASSIFICATION [LR03]

Phase	Robust positioning	Ad-hoc positioning	N -hop multilateration
1. Distance	DV-HOP	Euclidean	Sum-dist
2. Position	Lateralation	Lateralation	Min-max
3. Refinement	Yes	No	Yes

II. RELATED WORK

The localization problem and its variants has attracted significant research interest in recent years. A general survey of the area and an overview of recent techniques can be found in [NN01] and [MFA07], respectively. The problem is also closely related to dimensionality reduction [RS00] and manifold learning [SR03] in which the objects/data come from a high dimensional space, and the goal is to compute a low-dimensional, neighbourhood preserving embeddings.

In the case when all pairwise distances are known, the coordinates can be derived by using a classical method known as multidimensional scaling (MDS) [BG05]. The underlying principle of the MDS is to convert distances into an inner product matrix, whose eigenvectors are the unknown coordinates. In the presence of noise, MDS tolerates errors gracefully due to the overdetermined nature of the solution. However, when most pairwise distances are missing, the problem of finding the unknown coordinates becomes more challenging. For centralized algorithms (where all the measurements are sent to a single processor and the estimated positions are computed) three types of practical solutions to the above problem have been proposed in the literature. The first group consists of algorithms that try first to estimate the missing entries of the distance matrix and then apply MDS to the reconstructed distance matrix to find the coordinates of the sensors. MDS-MAP, introduced in [SRZF03] and further studied in [SRZF04], can be mentioned as a well-known example of this class where it computes the shortest paths between all pairs of nodes in order to approximate the missing entries of the distance matrix. The algorithms in the second group mainly consider the sensor localization as a non-convex optimization problem and directly estimate the coordinates of sensors. A famous example of this type is a relaxation to semidefinite programming (SDP)[BY04]. In the third group, the problem is formulated through a stochastic optimization where the main technique used in these algorithms is the stimulated annealing, which is a generalization of the Monte Carlo method in combinatorial optimization [KMV06], [KM06].

Perhaps a more practical and interesting case is when there is no central infrastructure. [LR03] identifies a common three-phase structure of three, popular, distributed sensor-localization algorithms, namely robust positioning [SLR02], ad-hoc positioning [NN03] and N-hop multilateration [SPS03]. Table I illustrates the structure of these algorithms. In the first phase, nodes share information to collectively determine the distances from each of the nodes to a number of anchors. Anchors are special nodes with a priori knowledge of their own position in some global coordinate system. In the second phase, nodes determine their position based on the estimated distances to the anchors provided by the first phase and the known positions of the anchors. In the last phase, the initial estimated positions are iteratively refined. It is empirically demonstrated that these simple three-phase distributed sensor-localization algorithms are robust and energy-efficient [LR03]. However, depending on which method is used in each phase, there are different tradeoffs between localization accuracy, computation complexity and power requirements. In [NSB03], a distributed algorithm-called the Gradient algorithm- was proposed; it is similar to ad-hoc positioning [NN03] but uses a different method for estimating the average distance per hop.

Another distributed approach introduced in [IFMW04] is to pose the localization problem as an inference problem on a graphical model and solve it by using Nonparametric Belief Propagation (NBP). It is naturally a distributed procedure and produces both an estimate of sensor locations and a representation of the location uncertainties. The estimated uncertainty may subsequently be used to determine the reliability of each sensor's location estimate.

The performances of these practical algorithms are invariably measured through simulations and little is known about the theoretical analysis supporting their results. A few exceptions are in the following work. In [DJMI⁺06] the authors use matrix completion methods [Faz02] as a means to reconstruct the distance matrix. The main contribution of their paper is that they are able to provably localize the sensors up to a bounded error. However, their analysis is based on a number of strong assumptions. First, they assume that even far-away sensors have a non-zero probability of detecting their distances. Second, the algorithm explicitly requires the knowledge of detection probabilities between all pairs. Third, their theorem only works when the average degree of the network (i.e., the average number of nodes detected by each sensor) grows linearly with the number of sensors in the network.

Our first result, specifically the analysis of MDS-MAP, has a similar flavour as in [DJMI⁺06]. We provide a theoretical guarantee that backs up experimental results. We use shortest paths as our primary guess for the missing entries in the distance matrix and apply MDS to find the topology of the network. In contrast to [DJMI⁺06], we require weaker assumptions for our results. More specifically, we assume that only neighbouring sensors have information about each other and that only connectivity information

is known. Furthermore, the knowledge of detection probabilities plays no role in our analysis or the algorithm. And last, in our analysis we assume that the average degree grows logarithmically-not linearly-with the number of sensors, which results in needing many less revealed entries in the distance matrix. In particular, the last condition is quite realistic: If the average degree grows any slower then the network is not even connected (more on this issue in Section IV-C). As the shortest paths algorithm works for both range-free and range-aware cases, our analysis includes both and provides the first error bounds on the performance of MDS-MAP.

Of particular interest are the two new results on the performance of sensor localization algorithms. In [JM11], Javanmard et al. proposes a new reconstruction algorithm based on semidefinite programming where they could establish lower and upper bounds on the reconstruction errors of their algorithm. Similarly, in [KOPV10], due to new advances in matrix completion methods [CR08], the authors analyse the performance of OptSpace [KM10], a novel matrix completion algorithm, in localizing the sensors. Interestingly, they did not need to adhere to the assumptions made by [DJMI⁺06]. However, they have a restrictive assumption about the topology of the network, specifically, sensors are scattered inside an annulus.

All the above analytical results crucially rely on the fact that there is a central processor with access to the inter-sensor distance measurements. However, as we have mentioned earlier, centralized algorithms suffer from the scalability problem and require higher computational complexity. Hence, a distributed algorithm with similar a performance bound is desirable. In our second result, we analyse the reconstruction error of a distributed sensor localization algorithm. To the best of our knowledge we show for the first time that HOP-TERRAIN, introduced in [SLR02], achieves a bounded error when only local connectivity information is given.

Finally, one of the fundamental challenges in localization problem is whether, given a set of measurements, the sensor network is uniquely localizable or not. In the noiseless setting where all the measurements are accurate, it was shown that the correct notion through which we can answer this question is the global rigidity [MWY06], a property that is easy to check (a thorough discussion of global rigidity and its implications for the sensor localization problem is given in [GHDT10]). However, finding such a unique solution is NP-hard [JJ05]. In the case of noisy distance measurements very little is known in this area. For instance, we do not know the fundamental limits for sensor localization algorithms or whether there are any algorithms with proven guarantees. From this point of view, our results narrow the gap between the algorithmic aspect of sensor localization and the theoretical one. In particular, we show that even in the presence of noise, the MDS-MAP and HOP-TERRAIN algorithms can localize

the nodes within a bounded error.

The organization of this paper is as follows. In Section III we introduce the model and the notation used in our work. In Section IV we describe the MDS-MAP and HOP-TERRAIN algorithms and their common features. Our results are stated in Section V where we provide their proofs in Section VI. Finally, we conclude in Section VII.

III. MODEL DEFINITION

Before discussing the centralized and distributed localization algorithms in detail, we define the mathematical model considered in this work. First, we assume that we have no fine control over the placement of the sensors that we call the *unknown nodes* (e.g., the nodes are dropped from an airplane). Formally, we assume that n nodes are placed uniformly at random in a d -dimensional cube $[0, 1]^d$.

Additionally, we assume that there are m special sensors, which we call *anchors*, with a priori knowledge of their own positions in some global coordinate. In practice, it is reasonable to assume that we have some control over the position of anchors. Basically, anchors are the nodes that are planted on the field before any positioning takes place.

Let $V_a = \{1, \dots, m\}$ denote the set of m vertices corresponding to the anchors and $V_u = \{m + 1, \dots, m + n\}$ the set of n vertices corresponding to the unknown nodes. We use x_i to denote the random position of the node i and X to denote the $n \times d$ position matrix where the i -th row corresponds to x_i .

In positioning applications, due to attenuation and power constraints, only measurements between close-by nodes are available. As a result, the pairwise distance measurements can be represented by a random geometric graph $G(n+m, R) = (V, E, P)$, where $V = V_u \cup V_a$, $E \subseteq V \times V$ is a set of undirected edges that connect pairs of sensors that are close to each other, and $P : E \rightarrow \mathbb{R}^+$ is a non-negative real-valued function. The function P is a mapping from a pair of connected nodes $(i, j) \in E$ to a distance measurement between i and j .

A common model for this random geometric graph is the disc model where node i and j are connected if the Euclidean distance $d_{i,j} \equiv \|x_i - x_j\|$ is less than or equal to a positive radio range R . In formulae,

$$(i, j) \in E \Leftrightarrow d_{i,j} \leq R.$$

As mentioned earlier, there are a variety of ways to measure the connectivity between two nodes, including time difference of arrival and RF received-signal strength (also called RF ranging). Due to limited resources, in all of the mentioned solutions there is a probability of non-detection (or completely wrong estimation). Think of RF ranging in the presence of an obstacle or in the (frequent) case of multiple

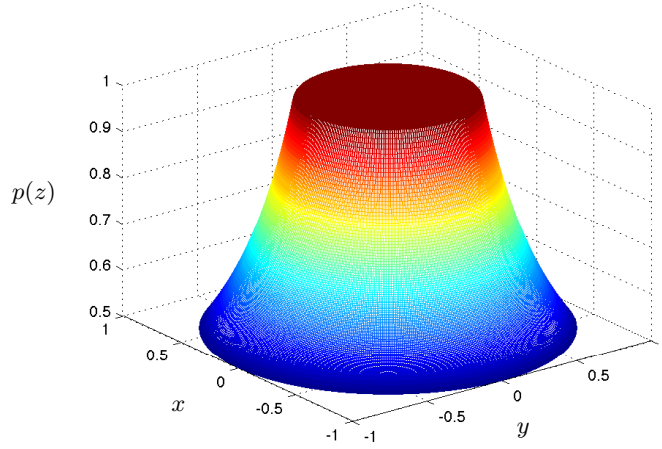


Fig. 1. This plot shows how the probability of detection changes as the distance between two sensor changes.

paths. Depending on the acquisition mechanism, this may result in the absence of measurement or in incoherent measurements.

Throughout this paper, to model this failure of detection, we assume that two nodes can detect each other with a probability that only depends on the distance $d_{i,j}$. Namely, $(i, j) \in E$ with probability $p(d_{i,j})$ if $d_{i,j} \leq R$. The detection probability $p(\cdot) : [0, R] \rightarrow [0, 1]$ is a non-increasing function of the distance. We consider a simple function parameterized by two scalar values $\alpha \in (0, 1]$ and $\beta \in [0, 3)$:

$$p(z) = \min \left(1, \alpha \left(\frac{z}{R} \right)^{-\beta} \right), \quad (1)$$

for $\alpha \in (0, 1]$ and $\beta \in [0, d)$. Note that this includes the disc model with perfect detection as a special case (i.e., $\alpha = 1, \beta = 0$).

To each edge $(i, j) \in E$, we associate the distance measurement $P_{i,j}$ between sensors i and j . In an ideal case, we have exact distance measurements available for those pairs in E . This is called the *range-based model* or the *range-aware model*. In formulae,

$$P_{i,j} = \begin{cases} d_{i,j} & \text{if } (i, j) \in E, \\ * & \text{otherwise,} \end{cases}$$

where a $*$ denotes that the distance measurement is unavailable.

In this paper, we assume that we are given only network connectivity information and no distance

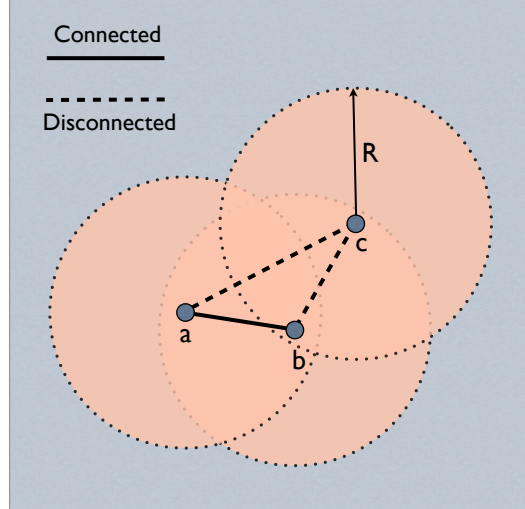


Fig. 2. This example shows the model we consider in this work. Nodes a and b are connected since they are within radio range R from each other. Even though the similar situation presents for b and c , they are not connected due to detection failure. Finally, nodes a and c are not connected because they are far apart.

information. This is known as the *connectivity-based model* or the range-free model. More formally,

$$P_{i,j} = \begin{cases} 1 & \text{if } (i,j) \in E, \\ * & \text{otherwise.} \end{cases}$$

In the following, let D denote the $n \times n$ squared distance matrix where $D_{i,j} = d_{i,j}^2$. By definition,

$$D = a\mathbf{1}_n^T + \mathbf{1}_n a^T - 2XX^T,$$

where $a \in \mathbb{R}^n$ is a vector with $a_i = \|x_i\|^2$ and $\mathbf{1}_n$ is the all ones vector. As D is a sum of two rank-1 matrices and a rank- d matrix, its rank is at most $d + 2$.

IV. ALGORITHMS

In general, there are two solutions to the localization problem: a relative map and an absolute map. A relative map is a configuration of sensors that have the same neighbor relationships as the underlying graph G . In the following we use the terms configuration, embedding, and relative map interchangeably. An absolute map, on the other hand, determines the absolute geographic coordinates of all sensors. In this paper our objective is two-fold. First, we present the centralized algorithm MDS-MAP, that finds a configuration that best fits the proximity measurements. Then, we discuss its distributed version HOP-TERRAIN where its goal is for each sensor to find its absolute position. For both, we provide analytical bounds on the error between the estimated configuration and the correct configuration.

TABLE II
SUMMARY OF NOTATION.

n	number of unknown sensors	V_u	set of unknown nodes
m	number of anchors	V_a	set of anchors
R	communication range	$\mathbb{1}_n$	all ones vector of size n
$P_{i,j}$	distance measurements	\hat{D}	estimated squared distance matrix
$d_{i,j}$	Pairwise distance between nodes i and j	$\mathbb{I}_{n \times n}$	$n \times n$ identity matrix
x_i	position of node i	\hat{x}_i	estimated position of node i
p	detection probability	X	positions matrix
d	dimension	\hat{X}	estimated positions matrix
D	squared distance matrix	$\hat{d}_{i,j}$	shortest path between node i and j
$O(d)$	orthogonal group of $d \times d$ matrices	$\ \cdot\ _F$	Frobenius norm
$\langle A, B \rangle$	Frobenius inner product	$\ \cdot\ _2$	spectral norm

A. Centralized Positioning Algorithm: MDS-MAP

For the centralized positioning algorithm, we assume that there is no anchor node in the system, namely, $V_a = \emptyset$. We define a set of random positions of n sensors $\mathcal{X} = \{x_1, \dots, x_n\}$. MDS-MAP consists of two steps:

Algorithm : MDS-MAP [SRZF03]

Input: dimension d , graph $G = (V, E, P)$

- 1: Compute the shortest paths, and let \hat{D} be the squared shortest paths matrix;
 - 2: Apply MDS to \hat{D} , and let \hat{X} be the output.
-

Shortest paths. The shortest path between nodes i and j in graph $G = (V, E, P)$ is defined as a path between two nodes such that the sum of the proximity measures of its constituent edges is minimized. Let $\hat{d}_{i,j}$ be the computed shortest path between node i and j . Then, the squared shortest paths matrix $\hat{D} \in \mathbb{R}^{n \times n}$ is defined as $\hat{D}_{ij} = \hat{d}_{i,j}^2$ for $i \neq j$, and 0 for $i = j$.

Multidimensional scaling. In step 2, we apply the MDS to \hat{D} to get a good estimate of X , specifically, we compute $\hat{X} = \text{MDS}_d(\hat{D})$. Multidimensional scaling (MDS) refers to a set of statistical techniques used in finding the configuration of objects in a low dimensional space such that the measured pairwise distances are preserved [BG05]. It is often used for a visual representation of the proximities between a set of items. For example, given a matrix of perceived similarities or dissimilarities between n items, MDS geometrically places each of those items in a low dimensional space such that the items that are similar are placed close to each other. Formally, MDS finds a lower dimensional embedding \hat{x}_i s that

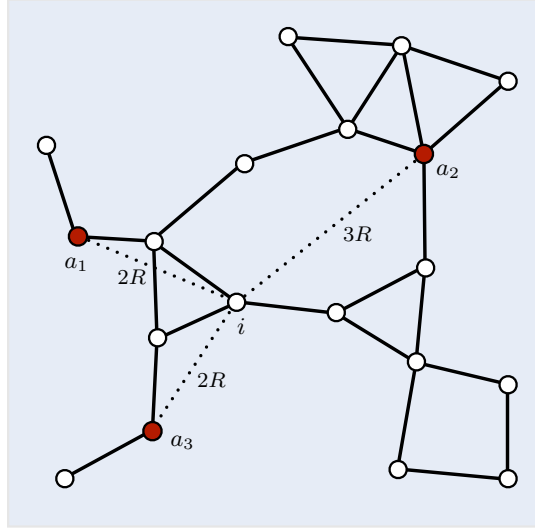


Fig. 3. The shortest path between two nodes is defined in terms of the minimum number of hops multiplied by the radio range R . For instance, the shortest path between i and a_1 is $2R$.

minimize the *stress* defined as

$$\text{stress} \doteq \sqrt{\frac{\sum_{i \neq j} (f(d_{i,j}) - \hat{d}_{i,j})^2}{\sum_{i \neq j} \hat{d}_{i,j}^2}},$$

where $d_{i,j}$ is the input similarity (or dissimilarity), $\hat{d}_{i,j} = \|\hat{x}_i - \hat{x}_j\|$ is the Euclidean distance in the lower dimensional embedding, and $f(\cdot)$ is some function on the input data. When MDS perfectly embeds the input data, we will have $f(d_{i,j}) = \hat{d}_{i,j}$ and the stress is zero.

In this chapter we use what is called the classic metric MDS (we refer the interested reader to [CC01], for the definition of other types of MDS algorithms, for instance non-metric MDS, replicated MDS, and weighted MDS). In classic metric MDS, $f(\cdot)$ is the identity function and the input dissimilarities correspond to the Euclidean distances such that $d_{i,j} = \|x_i - x_j\|$ for some lower dimensional embedding $\{x_i\}$. Further, when all the dissimilarities (or pairwise distances) are measured without error, the following spectral method correctly recovers the lower dimensional embedding up to a rigid motion.

Algorithm : Classic Metric MDS [SRZF03]

Input: dimension d , estimated distance matrix M

- 1: Compute $(-1/2)MLL$,
where $L = \mathbb{I}_n - (1/n)\mathbb{1}_n\mathbb{1}_n^T$;
 - 2: Compute the best rank- d approximation $U_d\Sigma_dU_d^T$
of $(-1/2)MLL$;
 - 3: Return $\text{MDS}_d(M) \equiv U_d\Sigma_d^{1/2}$.
-

This algorithm has been frequently used in positioning applications; and in the future, whenever we say MDS we refer to the above algorithm. Let L be an $n \times n$ symmetric matrix such that

$$L = \mathbb{I}_n - (1/n)\mathbb{1}_n\mathbb{1}_n^T,$$

where $\mathbb{1}_n \in \mathbb{R}^n$ is the all ones vector and \mathbb{I}_n is the $n \times n$ identity matrix. Let $\text{MDS}_d(D)$ denote the $n \times d$ matrix returned by MDS when applied to the squared distance matrix D . Then, in formula, given the singular value decomposition (SVD) of a symmetric and positive definite matrix $(-1/2)LDL$ as $(-1/2)LDL = U\Sigma U^T$,

$$\text{MDS}_d(D) \equiv U_d\Sigma_d^{1/2},$$

where U_d denotes the $n \times d$ left singular matrix that corresponds to the d largest singular values and Σ_d denotes the $d \times d$ diagonal matrix with d largest singular values in the diagonal. This is also known as the MDSLOCALIZE algorithm in [DJMI⁺06]. Note that as the columns of U are orthogonal to $\mathbb{1}_n$ by construction, it follow that $L \cdot \text{MDS}_d(D) = \text{MDS}_d(D)$.

It can be easily shown that when MDS is applied to the correct squared distance matrix without noise, the configuration of sensors are exactly recovered [DJMI⁺06]. This follows from the following equality

$$-(1/2)LDL = LXX^TL. \quad (2)$$

Note that we only obtain the configuration and not the absolute positions, in the sense that $\text{MDS}_d(D)$ is one version of infinitely many solutions that matches the distance measurements D . Therefore there are multiple incidents of X that result in the same D . We introduce a formal definition of rigid transformation and related terms.

We denote by $O(d)$ the orthogonal group of $d \times d$ matrices. A set of sensor positions $Y \in \mathbb{R}^{n \times d}$ is a rigid transformation of X , if there exists a d -dimensional shift vector s and an orthogonal matrix $Q \in O(d)$ such that $Y = XQ + \mathbb{1}_n s^T$. Here Y should be interpreted as a result of first rotating (and/or

reflecting) sensors in position X by Q and then adding a shift by s . Similarly, when we say two position matrices X and Y are equal up to a rigid transformation, we mean that there exists a rotation Q and a shift s such that $Y = XQ + \mathbb{1}_n s^T$. Also, we say a function $f(X)$ is *invariant* under rigid transformation if and only if for all X and Y that are equal up to a rigid transformation we have $f(X) = f(Y)$. Under these definitions, it is clear that D is invariant under rigid transformation, as for all (i, j) ,

$$D_{ij} = \|x_i - x_j\|^2 = \|(x_i Q + s^T) - (x_j Q + s^T)\|^2,$$

for any $Q \in O(d)$ and $s \in \mathbb{R}^d$.

Although MDS works perfectly when D is available, in practice not all proximity measurements are available because of the limited radio range R . This is why, in the first step, we estimated the unavailable entries of D by finding the shortest path between disconnected nodes.

B. Distributed Positioning Algorithm: HOP-TERRAIN

Recall that HOP-TERRAIN is a distributed algorithm that aims at finding the global map. Notice that in order to fix the global coordinate system in a d dimensional space, we need to know the positions of at least $d + 1$ nodes. As we defined before, these nodes whose global positions are known are called anchors. In this section we assume that we have m anchors in total, i.e., $V_a = \{1, 2, \dots, m\}$. Based on the robust positioning algorithm introduced in [SLR02], the distributed sensor localization algorithm consists of two steps :

Algorithm : HOP-TERRAIN[SLR02]

- 1: Each node i computes the shortest paths $\{\hat{d}_{i,a} : a \in V_a\}$ between itself and the anchors;
 - 2: Each node i derives an estimated position \hat{x}_i by triangulation with a least squares method.
-

Distributed shortest paths: Similarly to MDS-MAP, the first step is about finding the shortest path. The difference is that in the first step each of the unknown nodes only estimates the distances between itself and the anchors. These approximate distances will be used in the next triangulation step to derive an estimated position. In other words, the shortest path between an unknown node i and an anchor a in the graph G provides an estimate for the Euclidean distance $d_{i,a} = \|x_i - x_a\|$.

We denote by $\hat{d}_{i,a}$ the computed shortest path and this provides the initial estimate for the distance between the node i and the anchor a . When only the connectivity information is available and the corresponding graph $G = (V, E, P)$ is defined as in the *connectivity-based model*, the shortest path $\hat{d}_{i,a}$

is equivalent to the minimum number of hops between a node i and an anchor a multiplied by the radio range R .

In order to find the minimum number of hops from an unknown node $i \in V_u$ to an anchors $a \in V_a$ in a distributed way, we use a method similar to DV-HOP [NN03]. Each unknown node maintains a table $\{x_a, h_a\}$ that is initially empty, where $x_a \in \mathbb{R}^d$ refers to the position of the anchor a and h_a to the number of hops from the unknown node to the anchor a . First, each of the anchors initiate a broadcast containing its known location and a hop count of one. All of the one-hop neighbors surrounding the anchor, on receiving this broadcast, record the anchor's position and a hop count of one, and then broadcast the anchor's known position and a hop count of two. From then on, whenever a node receives a broadcast, it does one of the two things. If the broadcast refers to an anchor that is already in the record and the hop count is larger than or equal to what is recorded, then the node does nothing. Otherwise, if the broadcast refers to an anchor that is new or has a hop count that is smaller, the node updates its table with this new information on its memory and broadcasts the new information after incrementing the hop count by one.

To estimate the distances between the node and the anchors, when every node has computed the hop count to all the anchors, the number of hops is multiplied by the radio range R to estimate the distances between the node and the anchors. Note that to begin triangulation, not all the hop counts to all the anchors are necessary. A node can start triangulation as soon as it has estimated distances to $d + 1$ anchors. There is an obvious trade-off between the number of communications and their performance.

The above step of computing the minimum number of hops is the same distributed algorithm as described in DV-HOP. However, one difference is that instead of multiplying the number of hops by a fixed radio range R , in DV-HOP, the number of hops is multiplied by an average hop distance. The average hop distance is computed from the known pairwise distances between anchors and the number of hops between the anchors. although numerical simulations show that the average hop distance provides a better estimate, the difference between the computed average hop distance and the radio range R becomes negligible as n grows large.

Triangulation using least squares. In the second step, each unknown node $i \in V_u$ uses a set of estimated distances $\{\hat{d}_{i,a} : a \in V_a\}$ together with the known positions of the anchors, to perform a triangulation. The resulting estimated position is denoted by \hat{x}_i . For each node, the triangulation consists in solving a single instance of a least squares problem ($Ax = b$) and this process is known as Lateration [SRB01], [LR03].

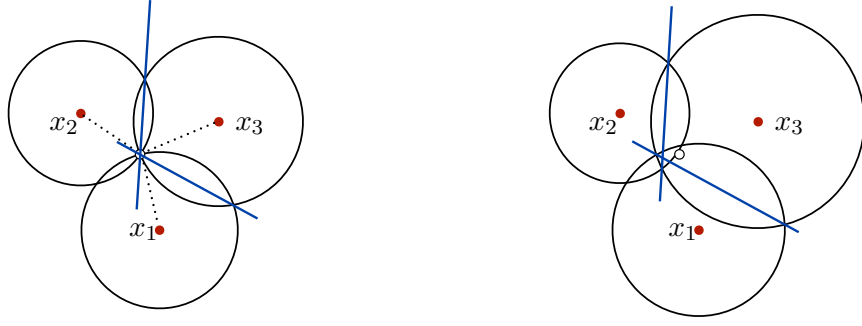


Fig. 4. Multilateration with exact distance measurements (left) and with approximate distance measurements (right). Three solid circles denote the anchors (red) and the white circle denotes the unknown nodes. The intersection of the blue lines corresponds to the solution of multilateration.

For an unknown node i , the position vector x_i and the anchor positions $\{x_a : a \in \{1, \dots, m\}\}$ satisfy the following series of equations:

$$\begin{aligned} \|x_1 - x_i\|^2 &= d_{i,1}^2, \\ &\vdots \\ \|x_m - x_i\|^2 &= d_{i,m}^2. \end{aligned}$$

Geometrically, the above equalities simply say that the point x_i is the intersection point of m circles centred at x_1, x_2, \dots, x_m (see Figure 4). This set of equations can be linearised by subtracting each line from the next line.

$$\begin{aligned} \|x_2\|^2 - \|x_1\|^2 + 2(x_1 - x_2)^T x_i &= d_{i,2}^2 - d_{i,1}^2, \\ &\vdots \\ \|x_m\|^2 - \|x_{m-1}\|^2 + 2(x_{m-1} - x_m)^T x_i &= d_{i,m}^2 - d_{i,m-1}^2. \end{aligned}$$

By reordering the terms, we get a series of linear equations for node i in the form $Ax_i = b_0^{(i)}$, for $A \in \mathbb{R}^{(m-1) \times d}$ and $b \in \mathbb{R}^{m-1}$ defined as

$$A \equiv \begin{bmatrix} 2(x_1 - x_2)^T \\ \vdots \\ 2(x_{m-1} - x_m)^T \end{bmatrix},$$

$$b_0^{(i)} \equiv \begin{bmatrix} \|x_1\|^2 - \|x_2\|^2 + d_{i,2}^2 - d_{i,1}^2 \\ \vdots \\ \|x_{m-1}\|^2 - \|x_m\|^2 + d_{i,m}^2 - d_{i,m-1}^2 \end{bmatrix}.$$

Note that the matrix A does not depend on the particular unknown node i and all the entries are known accurately to all the nodes after the distributed shortest paths step. However, the vector $b_0^{(i)}$ is not available at node i , because $d_{i,a}$'s are not known. Hence we use an estimation $b^{(i)}$, that is defined from $b_0^{(i)}$ by replacing $d_{i,a}$ by $\hat{d}_{i,a}$ everywhere. Notice that $\hat{d}_{i,a} \geq d_{i,a}$. As a result, the circles centred at x_1, x_2, \dots, x_m have potentially larger radii. Therefore, the intersection between circles is no longer a single point, but rather a closed area. Then, finding the optimal estimation \hat{x}_i of x_i that minimizes the mean squared error is solved in a closed form using a standard least squares approach:

$$\hat{x}_i = (A^T A)^{-1} A^T b^{(i)}. \quad (3)$$

For bounded $d = o(1)$, a single least squares operation has complexity $O(m)$, and applying it n times results in the overall complexity of $O(nm)$. No communication between the nodes is necessary for this step.

C. Stretch Factor: Euclidean Distance versus Shortest Path

In general when the graph G is not connected, the localization problem is not well defined. In fact, there are multiple configurations resulting in the same observed proximity measures. For instance if graph G consists of two disconnected components, they can be placed in possibly infinitely different ways with respect to each other without violating any constraints imposed by G . For this reason we restrict our attention to the case where G is connected.

In this work, we are interested in a scalable system of n unknown nodes for a large value of n . As n grows, it is reasonable to assume that the average number of connected neighbours for each node should stay constant. This happens, in our model, if we chose the radio range $R = C/n^{1/d}$. However, in the unit square, assuming sensor positions are drawn uniformly, the random geometric graph is connected, with

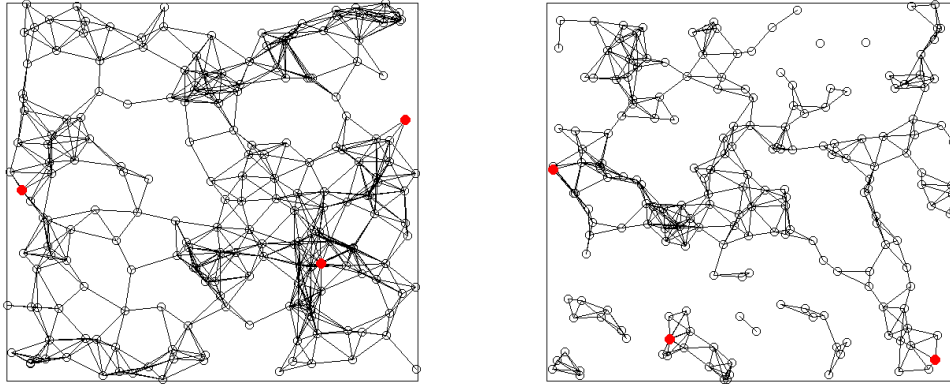


Fig. 5. The red vertices indicate the anchors. Under the right scaling of the radio range R , the graph stays connected (left figure) whereas otherwise there will be nodes without any means of communication to others (right graph).

high probability, if $\pi R^2 > (\log n + c_n)/n$ for $c_n \rightarrow \infty$ [GK98]. A similar condition can be derived for generic d -dimensions as $C_d R^d > (\log n + c_n)/n$, where $C_d \leq \pi$ is a constant that depends on d . Moreover, in case $C_d R^d < (\log n + c_n)/n$, not only the graph is not connected, there will be *isolated nodes* with high probability. Since isolated nodes cannot communicate with other sensors, there is no way to find their shortest paths to other nodes. Consequently, both MDS-MAP and HOP-TERRAIN algorithms will be in trouble (see Figure 5). Hence, instead of $R = C/n^{1/d}$, we focus in the regime where the average number of connected neighbors is slowly increasing with n . Let R_{critical} be the critical detection range where the resulting graph starts to be connected. Then we are interested in the regime $R = CR_{\text{critical}}$, for some positive constant $C \geq 1$ such that the graph stays connected with high probability.

In our analysis, the key observation and the crux of the argument is to show that the shortest-path estimate is guaranteed to be arbitrarily close to the correct distance for large enough radio range R and large enough n . Once we proved this, we can then show that the MDS step (equivalently, lateration) finds almost correctly the relative (equivalently, global) position of the sensors. We demonstrate how the error in estimating the Euclidean distance will be reflected on the position estimation. The precise statements are given in Section V.

We have already discussed the complexity of MDS and lateration steps. To complete our discussion we need to evaluate the complexity of finding the shortest path. In the MDS-MAP algorithm we require that all-pairs shortest paths be found. This problem has an efficient algorithm whose complexity is

$O(n^2 \log n + n|E|)$ [Joh77]. For $R = C(\log n/n)^{1/d}$ with constant C , the graph is sparse with $|E| = O(n \log n)$, whence the complexity is $O(n^2 \log n)$. Contrary to MDS-MAP, in HOP-TERRAIN we must only compute the shortest paths between the unknown nodes and the anchors. This distributed shortest paths algorithm can be done efficiently with total complexity of $O(nm)$.

V. MAIN RESULTS

In this section we present our main results regarding the performance of MDS-MAP and HOP-TERRAIN algorithms.

A. MDS-MAP

Our first result establishes an upper bound on the error achieved by MDS-MAP when we have only the connectivity information as in the case of the *connectivity-based model*.

Let \hat{X} denote an $n \times d$ estimation for X with an estimated position for node i in the i th row. Then, we need to define a metric for the distance between the original position matrix X and the estimation \hat{X} , which is invariant under rigid transformation of X or \hat{X} .

Define $L \equiv \mathbb{I}_n - (1/n)\mathbb{1}_n\mathbb{1}_n^T$ as in the MDS algorithm. L is an $n \times n$ rank $n-1$ symmetric matrix, which eliminates the contributions of the translation, in the sense that $LX = L(X + \mathbb{1}s^T)$ for all $s \in \mathbb{R}^d$. Note that L has the following nice properties:

- 1) LXX^TL is invariant under rigid transformation.
- 2) $LXX^TL = L\hat{X}\hat{X}^TL$ implies that X and \hat{X} are equal up to a rigid transformation.

This naturally defines the following distance between X and \hat{X} .

$$d_{\text{inv}}(X, \hat{X}) = \frac{1}{n} \|LXX^TL - L\hat{X}\hat{X}^TL\|_F, \quad (4)$$

where $\|A\|_F = (\sum_{i,j} A_{ij}^2)^{1/2}$ denotes the Frobenius norm. Notice that the factor $(1/n)$ corresponds to the usual normalization by the number of entries in the summation. Indeed this distance is invariant to rigid transformation of X and \hat{X} . Furthermore, $d_{\text{inv}}(X, \hat{X}) = 0$ implies that X and \hat{X} are equal up to a rigid transformation. With this metric, our main result establishes an upper bound on the resulting error. The proof of this theorem is provided in Section VI. We define

$$R_{\text{MDS}} \equiv 32 \left(\frac{12 \log n}{\alpha(n-2)} \right)^{\frac{1}{d}}. \quad (5)$$

Theorem 5.1 (connectivity-based model): Assume n nodes are distributed uniformly at random in the $[0, 1]^d$ hypercube, for a bounded dimension $d \in \{2, 3\}$. For a positive radio range R and detection

probability p defined in (1), we are given the connectivity information of the nodes according to the range-free model with probabilistic detection. Then, with a probability larger than $1 - 1/n^4$, the distance between the estimate \hat{X} produced by MDS-MAP and the correct position matrix X is bounded by

$$d_{\text{inv}}(X, \hat{X}) \leq \frac{R_{\text{MDS}}}{R} + 20R, \quad (6)$$

for $R > (1/\alpha)^{1/d} R_{\text{MDS}}$, where $d_{\text{inv}}(\cdot)$ is defined in (4) and R_{MDS} in (5).

The proof is provided in Section VI. The following corollary trivially follows, as for each $(i, j) \in E$, we have $d_{i,j} \leq R$.

Corollary 5.2 (range-based model): Under the hypotheses of Theorem 5.1 and in the case of range-based model, with high probability

$$d_{\text{inv}}(X, \hat{X}) \leq \frac{R_{\text{MDS}}}{R} + 20R.$$

As described in the previous section, we are interested in the regime where $R = C(\log n/n)^{1/d}$ for some constant C . Given a small positive constant δ , this implies that MDS-MAP is guaranteed to produce estimated positions that satisfy $d_{\text{inv}}(X, \hat{X}) \leq \delta$ with a large enough constant C and a large enough n .

When α is fixed and $R = C(\log n/n)^{1/d}$ for some positive parameter C , the error bound in (6) becomes

$$d_{\text{inv}}(X, \hat{X}) \leq \frac{C_1}{C\alpha^{1/d}} + C_2 C \left(\frac{\log n}{n} \right)^{1/d},$$

for some numerical constants C_1 and C_2 . The first term is inversely proportional to C and $\alpha^{1/d}$ and is independent of n , whereas the second term is linearly dependent on C and vanishes as n grows large. This is illustrated in Figure 6, which shows numerical simulations with n sensors randomly distributed in the 2-dimensional unit square. Notice that the resulting error is inversely proportional to α and independent of β .

Remark 5.3: Even though the upper bounds for both range-free and range-based models have the same form, there is a slight difference between their behaviours as R grows. In the range-free case, up to some point, the performance of MDS-MAP improves as R increases. This is due to the fact that the first and second terms go in opposite directions as a function of R . However, In the range-based case, as R increases, we obtain a more accurate estimate of the the Euclidean distance. As a result, once the radio range increases, the resulting error of MDS-MAP decreases and we do not see the contribution of the second term. This phenomenon is illustrated in Figure 7.

Using the above theorem, we can further show that there is a linear transformation $S \in \mathbb{R}^{d \times d}$, such that when applied to the estimations, we get a similar bound in the Frobenius norm of the error in the positions.

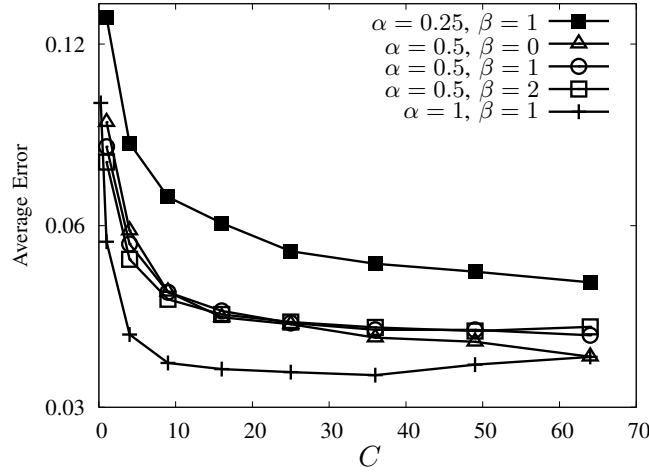


Fig. 6. Average distance between the correct topology X and the estimation \hat{X} using MDS-MAP as a function of C where the radio range is $R = C\sqrt{\log n/n}$. The $n = 1,000$ sensors are distributed randomly on a unit square under range-free model. Various values of α and β are used where two nodes at distance r are detected with probability $p(r) = \min\{1, \alpha(R/r)^\beta\}$.

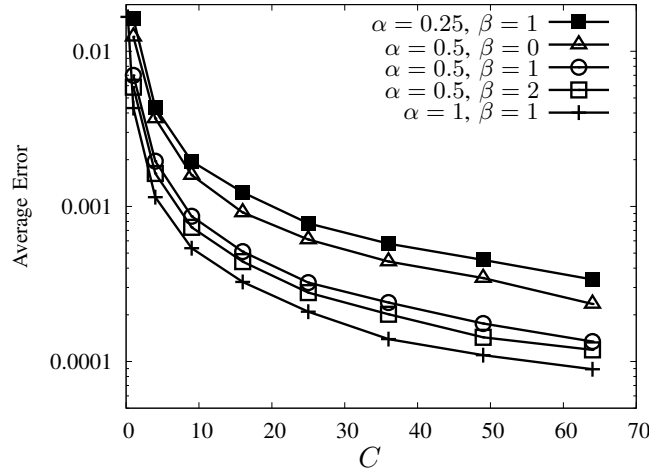


Fig. 7. Average error of MDS-MAP under the range-based model.

Theorem 5.4: Under the hypotheses of Theorem 5.1, with high probability

$$\min_{S \in \mathbb{R}^{d \times d}} \frac{1}{\sqrt{n}} \|LX - L\hat{X}S\| \leq \sqrt{6} \left(\frac{R_{\text{MDS}}}{R} + 20R \right)$$

Remark 5.5: Note that although for the sake of simplicity, we focus on $[0, 1]^d$ hypercube; our analysis easily generalizes to any bounded convex set and homogeneous Poisson process model with density $\rho = n$. The homogeneous Poisson process model is characterized by the probability that there are exactly k nodes appearing in any region with volume A : $\mathbb{P}(k_A = k) = \frac{(\rho A)^k}{k!} e^{-\rho A}$. Here, k_A is a random variable

defined as the number of nodes in a region of volume A .

Remark 5.6: To simplify calculations, we assumed that d is either 2 or 3. However, the analysis easily applies to general d and only the constant in the bound (6) would change as long as $d = O(1)$.

In what follows we investigate an important question whether similar performance guarantees, as in MDS-MAP, can be obtained in a distributed setting. In particular, we analyze the performance of the HOP-TERRAIN algorithm. As we have already stressed, this algorithm can be seen as a distributed version of the MDS-MAP algorithm. In particular, we show that when only connectivity information is available, for every unknown node. The Euclidean distance between the estimate and the correct position can be bounded very similarly to Theorem 5.1.

B. HOP-TERRAIN

Our second result establishes that HOP-TERRAIN [SLR02] achieves an arbitrarily small error for a radio range $R = C(\log n/n)^{1/d}$ with a large enough constant C , when we have only the connectivity information as in the case of the *connectivity-based model*. The same bound holds immediately for the *range-based model*, when we have an approximate measurements for the distances, and the same algorithm can be applied without any modification. to compute better estimates for the actual distances between the unknown nodes and the anchors, the extra information can be readily incorporated into the algorithm. We define

$$R_{\text{HOP}} \equiv 12 \left(\frac{12 \log n}{\alpha(n-2)} \right)^{\frac{1}{d}}. \quad (7)$$

Theorem 5.7: Assume n sensors and m anchors are distributed uniformly at random in the $[0, 1]^d$ hypercube for a bounded dimension $d \in \{2, 3\}$. For a given radio range $R > (1/\alpha)^{1/d} R_{\text{HOP}}$, detection probability p defined in (1), and the number of anchors $m = \Omega(\log n)$, the following is true with probability at least $1 - 1/n^4$. For all unknown nodes $i \in V_u$, the Euclidean distance between the estimate \hat{x}_i given by HOP-TERRAIN and the correct position x_i is bounded by

$$\|x_i - \hat{x}_i\| \leq \frac{R_{\text{HOP}}}{R} + 24R. \quad (8)$$

The proof is provided in Section VI. As described in the previous section, we are interested in the regime where $R = C(\log n/n)^{1/d}$ for some constant C . Given a small positive constant δ , this implies that HOP-TERRAIN is guaranteed to produce estimated positions that satisfy $\|x_i - \hat{x}_i\| \leq \delta$ for all $i \in V_u$ with a large enough constant α and large enough n .

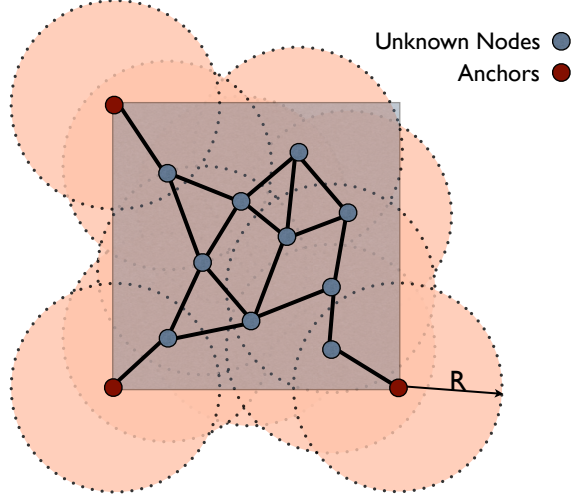


Fig. 8. Three anchors in fixed positions $([0, 0], [1, 0], [0, 1])$ for a two-dimensional sensor localization.

When the number of anchors is bounded and the positions of the anchors are chosen randomly, it is possible that, in the triangulation step, we get an ill-conditioned matrix $A^T A$, resulting in a large estimation error. This happens, for instance, if three anchors fall close to a line. However, as mentioned in the introduction, it is reasonable to assume that, for the anchors, the system designer has some control over where they are placed. In that case, the next remark shows that when the positions of anchors are properly chosen, only $d + 1$ anchors suffice to get a similar bound on the performance. Note that this is the minimum number of anchors necessary for triangulation. For simplicity we assume that one anchor is placed at the origin and d anchors are placed at positions corresponding to d -dimensional unit vectors. The position of the $d + 1$ anchors are $\{[0, \dots, 0], [1, 0, \dots, 0], [0, 1, 0, \dots, 0], [0, \dots, 0, 1]\}$. (see figure 8)

Theorem 5.8: Assume that n sensors are distributed uniformly at random in the $[0, 1]^d$ hypercube for a bounded dimension $d = \{2, 3\}$. Also, assume that there are $d + 1$ anchors, one of which is placed at the origin, and the position vectors of the d remaining anchors are the d -dimensional unit vectors. For a given radio range $R > (1/\alpha)^{1/d} R_{\text{HOP}}$ and detection probability p defined in (1) the following is true with probability at least $1 - 1/n^4$. For all unknown nodes $i \in V_u$, the Euclidean distance between the estimate \hat{x}_i given by HOP-TERRAIN and the correct position x_i is bounded by

$$\|x_i - \hat{x}_i\| \leq 2 \frac{R_{\text{HOP}}}{R} + 48R. \quad (9)$$

The proof is provided in Section VI.

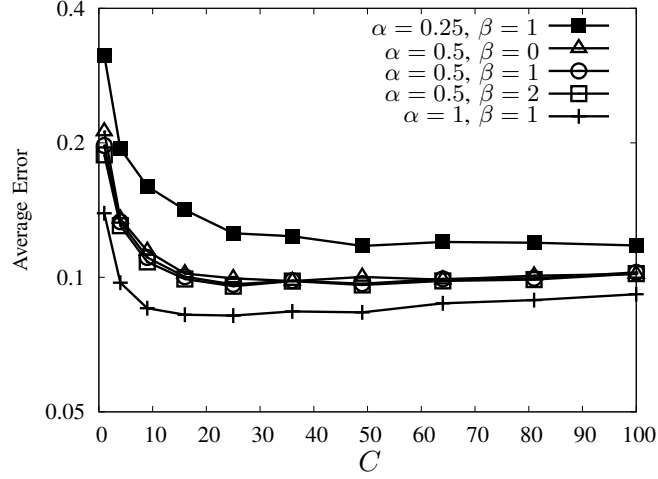


Fig. 9. Average distance between the correct position $\{x_i\}$ and estimation $\{\hat{x}_i\}$ using HOP-TERRAIN as a function of C , for $R = C\sqrt{\log n/n}$ with $n = 5,000$ sensors in the unit square under connectivity-based model. Two nodes at distance r detect each other with probability $p(r) = \min\{1, \alpha(R/r)^\beta\}$.

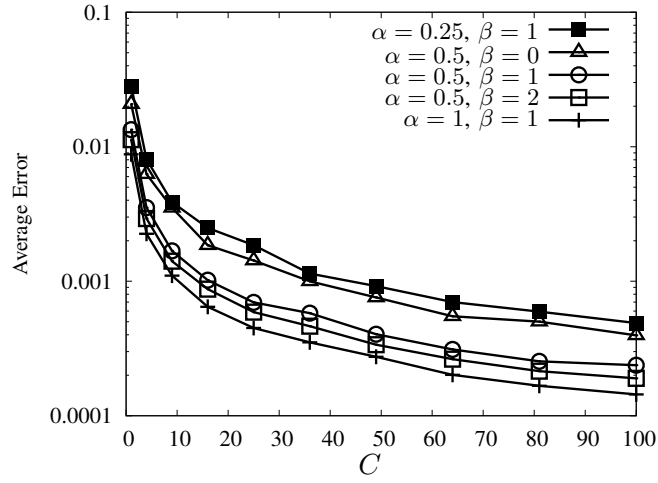


Fig. 10. Average error under range-based model.

Remark 5.9: There is nothing particular about the position of the anchors in unit vectors. Any $d + 1$ anchors in general position will give similar bounds. The only difference is that the constant term in the definition of R_{HOP} changes with the anchor positions.

Corollary 5.10 (range-based model): Under the hypothesis of Theorem 5.7 and in the range-based

model, with high probability

$$\|x_i - \hat{x}_i\| \leq \frac{R_{\text{HOP}}}{R} + 24R.$$

The similar result holds true when sensors are placed deterministically, specifically, under the hypothesis of Theorem 5.8, with high probability,

$$\|x_i - \hat{x}_i\| \leq 2\frac{R_{\text{HOP}}}{R} + 48R.$$

As it was the case for MDS-MAP, when $R = C(\log n/n)^{1/d}$ for some positive parameter C , the error bound in (9) is

$$\|x_i - \hat{x}_i\| \leq \frac{C_1}{C\alpha^{1/d}} + C_2C \left(\frac{\log n}{n} \right)^{1/d}$$

for some numerical constants C_1 and C_2 . The first term is inversely proportional to C and $\alpha^{1/d}$ and is independent of n , whereas the second term is linearly dependent in C and vanishes as n grows large. This is illustrated in Figure 9, which shows numerical simulations with $n = 5,000$ sensors randomly distributed in the 2-dimensional unit square. We compute the root mean squared error: $\{(1/n) \sum_{i=1}^n \|x_i - \hat{x}_i\|^2\}^{1/2}$.

Figure 11 shows a network consisting of $n = 200$ nodes placed randomly in the unit circle. The three anchors in fixed positions are displayed by solid blue circles. In this experiment the distance measurements are from the range-based model and the radio range is $\sqrt{0.8 \log n/n}$. Figure 12 shows the final estimated positions using HOP-TERRAIN. The circles represent the correct positions, and the solid lines represent the differences between the estimates and the correct positions. The average error in this example is 0.075.

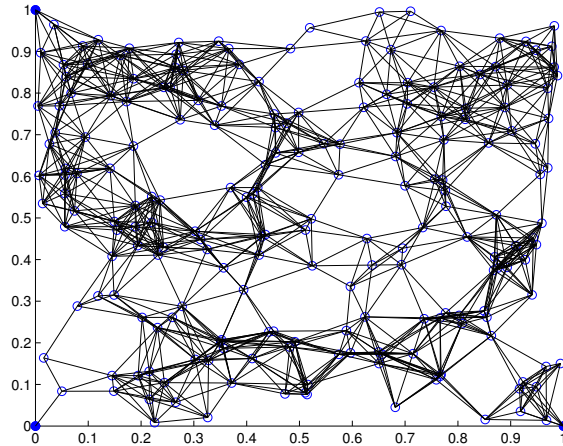


Fig. 11. 200 nodes randomly placed in the unit square and 3 anchors in fixed positions. The radio range is $R = \sqrt{0.8 * \log n/n}$.

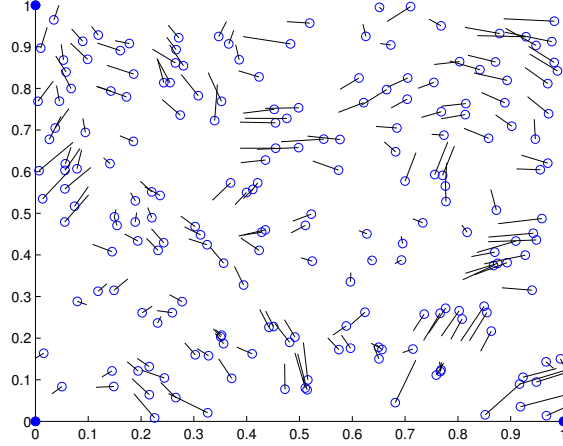


Fig. 12. Location estimation using HOP-TERRAIN.

VI. PROOF OF THE MAIN THEOREMS

A. Proof of Theorem 5.1

We start by bounding the distance $d_{\text{inv}}(X, \hat{X})$, as defined in Eq. (4), in terms of D and \hat{D} . Let $\|A\|_F = (\sum_{i,j} A_{ij}^2)^{1/2}$ denote the Frobenius norm of a matrix and $\|A\|_2 = \max_{\|x\|=1} \|Ax\|_2$ denote the spectral norm. Note that for a rank r matrix A we have

$$\|A\|_2 \leq \|A\|_F \leq r\|A\|_2.$$

Since $L(XX^T - \hat{X}\hat{X}^T)L$ has rank at most $2d$, it follows that

$$\|L(XX^T - \hat{X}\hat{X}^T)L\|_F \leq \sqrt{2d}\|L(XX^T - \hat{X}\hat{X}^T)L\|_2. \quad (10)$$

To bound the spectral norm, let $M = -(1/2)L\hat{D}L$. Then,

$$\begin{aligned} \|L(XX^T - \hat{X}\hat{X}^T)L\|_2 &\leq \|LXX^TL - M\|_2 + \|M - \hat{X}\hat{X}^T\|_2 \\ &\leq (1/2)\|L(-D + \hat{D})L\|_2 + (1/2)\|L(-\hat{D} + D)L\|_2 \\ &\leq \|\hat{D} - D\|_2, \end{aligned} \quad (11)$$

where in the first inequality we used the triangular inequality and the fact that $\hat{X} = L\hat{X}$. In the second inequality we used (2) and the fact that

$$\|M - \hat{X}\hat{X}^T\|_2 = \min_{A: \text{rank}(A) \leq d} \|M - A\|_2,$$

which follows from the definition of \hat{X} . From the definition of $\hat{X} = \text{MDS}_d(\hat{D})$, we know that $\hat{X}\hat{X}^T$ is the best rank- d approximation to M . Hence, $\hat{X}\hat{X}^T$ minimizes $\|M - A\|_2$ for any rank- d matrix A . Since the rank of $-(1/2)LDL$ is d , this implies

$$\|M - \hat{X}\hat{X}^T\|_2 \leq \|M + (1/2)LDL\|_2.$$

The inequality in (11) follows trivially from the observation that $\|L\|_2 = 1$.

Next, to bound $\|\hat{D} - D\|_2$, we use the following key result on the number of hops in graph G . The main idea is that, for sensors with uniformly random positions, the number of hops scaled by the radio range R provide estimates close to the correct distance. We define

$$\tilde{R} \equiv 2 \left(\frac{12 \log n}{\alpha(n-2)} \right)^{\frac{1}{d}}. \quad (12)$$

Lemma 6.1: (Bound on the distance estimation) Under the hypotheses of Theorem 5.1, with probability larger than $1 - 1/n^4$, for any pair of nodes $i \in V$ and $j \in V$, the number of hops between nodes i and j is bounded by

$$h_{i,j} \leq \left(1 + \frac{\tilde{R}}{R}\right) \frac{d_{i,j}}{R} + 2,$$

for $R > \max\{7\tilde{R}, (1/\alpha)^{1/d}\tilde{R}\}$.

The proof of this lemma is provided in Section VI-G. The distance estimate from the first step of MDS-MAP is $\hat{d}_{i,j} = Rh_{i,j}$. The following corollary gives a bound on the estimation error.

Corollary 6.2: Under the hypotheses of Lemma 6.1,

$$\hat{d}_{i,j}^2 - d_{i,j}^2 \leq \frac{30\tilde{R}}{14R} d_{i,j}^2 + 8R.$$

Proof: From Lemma 6.1, we know that

$$(Rh_{i,j})^2 - d_{i,j}^2 \leq \frac{2\tilde{R}}{R} \left(1 + \frac{\tilde{R}}{2R}\right) d_{i,j}^2 + 2R \left(1 + \frac{\tilde{R}}{R}\right) d_{i,j} + 4R^2.$$

The corollary follows from the assumption that $7\tilde{R} < R \leq 1$ and $d \leq 3$. ■

Define an error matrix $Z = \hat{D} - D$. Then by Corollary 6.2, Z is element-wise bounded by

$$0 \leq Z_{ij} \leq (30\tilde{R}/(14R))D_{ij} + 8R.$$

We can bound the spectral norm of Z as follows. Let u and v be the left and right singular vectors of the non-negative matrix Z , respectively. Then by Perron-Frobenius theorem, u and v are also non-negative.

It follows that

$$\begin{aligned}
\|\hat{D} - D\|_2 &= u^T Z v \\
&\leq (30\tilde{R}/(14R))u^T D v + (\mathbf{1}^T u)(\mathbf{1}^T v)8R \\
&\leq (30\tilde{R}/(14R))\|D\|_2 + 8Rn \\
&\leq (30\tilde{R}/(14R))dn + 8Rn.
\end{aligned} \tag{13}$$

The first inequality follows from the element-wise bound on Z and the non-negativity of u and v , and the second inequality follows from the definition of the spectral norm and the Cauchy-Schwarz inequality. In the last inequality, we used $\|D\|_2 \leq dn$, which follows from the fact that D is non-negative and element-wise bounded by d . Typically we are interested in the regime where $R = o(1)$, and by assumption we know that $R \geq \tilde{R}$ and $d \leq 3$. Therefore, the first term in (13) dominates the error. Substituting this bound on $\|\hat{D} - D\|_2$ in (11) proves the theorem.

B. Proof of Theorem 5.4

Using SVD we can write LX as $U_{n \times d} \Sigma_{d \times d} V_{d \times d}^T$ where $U^T U = \mathbb{I}_{d \times d}$, $V^T V = V V^T = \mathbb{I}_{d \times d}$ and Σ is a diagonal matrix. We also denote the *Frobenius inner product* between two matrices $A_{m \times n}$ and $B_{m \times n}$ by

$$\langle A, B \rangle \doteq \sum_{i,j} A_{i,j} B_{i,j}.$$

It is easy to show that

$$\langle A, B \rangle = \text{Tr}(A^T B) \leq \|A\|_F \|B\|_F.$$

In fact, this inner product induces the Frobenius norm definition. In particular, for an $m \times n$ matrix A we have

$$\|A\|_F = \sup_{B \in \mathbb{R}^{m \times n}, \|B\|_F \leq 1} \langle B, A \rangle.$$

Now, for $S = \hat{X}^T L U \Sigma^{-1} V^T$, we have

$$\begin{aligned}
\|LX - L\hat{X}S\|_F &= \sup_{B \in \mathbb{R}^{n \times d}, \|B\|_F \leq 1} \langle B, LX - L\hat{X}S \rangle \\
&= \sup_{B \in \mathbb{R}^{n \times d}, \|B\|_F \leq 1} \langle B, (LXV\Sigma U^T - L\hat{X}\hat{X}^T L)U\Sigma^{-1}V^T \rangle \\
&= \sup_{B \in \mathbb{R}^{n \times d}, \|B\|_F \leq 1} \langle BV\Sigma^{-1}U^T, LXX^T L - L\hat{X}\hat{X}^T L \rangle \\
&= \sup_{B \in \mathbb{R}^{n \times d}, \|B\|_F \leq 1} \|BV\Sigma^{-1}U^T\|_F \|LXX^T L - L\hat{X}\hat{X}^T L\|_F.
\end{aligned}$$

Using the fact $\|A\|_F = \text{Tr}(A^T A)$ and the cyclic property of the trace, i.e., $\text{Tr}(ABC) = \text{Tr}(BCA)$, we obtain

$$\|BV\Sigma^{-1}U^T\|_F = \text{Tr}(BV\Sigma^{-2}V^TB) \leq \sigma_{\min}^2 \|B\|_F^2,$$

where σ_{\min} is the smallest singular value of LX . It remains to show that $\sigma_{\min} \geq \sqrt{n/6}$ holds with high probability when nodes are placed uniformly at random. To this end we need to consider two facts. First, the singular values (and in particular the smallest singular value) are Lipschitz functions of the entries (See appendix). Second, we have $E(LX\hat{X}L) = (n/12)\mathbb{I}_{d \times d}$. By using concentration of measure for Lipschitz functions on bounded independent random variables, the result follows.

C. Proof of Theorem 5.7

In this section we provide the proofs of the theorems 5.7. Detailed proofs of the technical lemmas are provided in the following sections.

For an unknown node i , the estimation \hat{x}_i is given in Eq. (3).

$$\begin{aligned} \|x_i - \hat{x}_i\| &= \|(A^T A)^{-1} A^T b_0^{(i)} - (A^T A)^{-1} A^T b^{(i)}\| \\ &\leq \|(A^T A)^{-1} A^T\|_2 \|b_0^{(i)} - b^{(i)}\|, \end{aligned} \quad (14)$$

First, to bound $\|b_0^{(i)} - b^{(i)}\|$, we use Corollary 6.2. Since $d_{i,j}^2 \leq d$ for all i and j , we have

$$\begin{aligned} \|b_0^{(i)} - b^{(i)}\| &= \left(\sum_{k=1}^{m-1} (d_{i,k+1}^2 - d_{i,k}^2 - \hat{d}_{i,k+1}^2 + \hat{d}_{i,k}^2)^2 \right)^{1/2} \\ &\leq 2\sqrt{m-1} \left(\frac{30\tilde{R}}{14R} d + 8R \right), \end{aligned} \quad (15)$$

Next, to bound $\|(A^T A)^{-1} A^T\|_2$, we use the following lemma.

Lemma 6.3: Under the hypothesis of Theorem 5.7, the following is true. Assuming random anchor model in which $m = \Omega(\log n)$ anchors are chosen uniformly at random among n sensors. Then we have

$$\|(A^T A)^{-1} A^T\|_2 \leq \sqrt{\frac{3}{m-1}},$$

with high probability.

By assumption we know that $R \geq \tilde{R}$ and $d \leq 3$. By combining (14), (15) and Lemma 6.3 proves Theorems 5.7.

D. Proof of Theorem 5.8

In this section we provide the proof of Theorem 5.8. Detailed proofs of the technical lemmas are provided in the following sections.

Similarly to the proof of Theorem 5.7, for an unknown node i , and the estimate \hat{x}_i we have

$$\|x_i - \hat{x}_i\| \leq \|(A^T A)^{-1} A^T\|_2 \|b_0^{(i)} - b^{(i)}\|,$$

We have already bounded the expression $\|b_0^{(i)} - b^{(i)}\|$ in (15). To bound $\|(A^T A)^{-1} A^T\|_2$, we use the following lemma.

Lemma 6.4: Under the hypothesis of Theorem 5.8, the following are true. We assume a deterministic anchor model, where $m = d + 1$ anchors are placed on the positions

$$\begin{aligned} x_1 &= [1, 0, \dots, 0], \\ x_2 &= [0, 1, 0, \dots, 0], \\ &\vdots \\ x_d &= [0, 0, \dots, 0, 1], \\ x_{d+1} &= [0, 0, \dots, 0]. \end{aligned}$$

Then,

$$\|(A^T A)^{-1} A^T\|_2 \leq \frac{d}{2},$$

with high probability. This finishes the proof of Theorems 5.8.

E. Proof of Lemmas 6.3 (Random Model)

As it was the case in the proof of Lemma 6.4 in order to upper bound $\|(A^T A)^{-1} A\|_2$ we need to lower bound the smallest singular value of A . Let the symmetric matrix B be defined as $A^T A$. The diagonal entries of B can be written as

$$b_{i,i} = 4 \sum_{k=1}^{m-1} (x_{k,i} - x_{k+1,i})^2, \quad (16)$$

for $1 \leq i \leq d$ and the off-diagonal entries as

$$b_{i,j} = 4 \sum_{k=1}^{m-1} (x_{k,i} - x_{k+1,i})(x_{k,j} - x_{k+1,j}), \quad (17)$$

for $1 \leq i \neq j \leq d$ where $x_{k,i}$ is the i -th element of vector x_k . In the following lemmas, we show that with high probability, as m increases, the diagonal entries of B will all be of the order of m , i.e., $b_{i,i} = \Theta(m)$, and the off-diagonal entries will be bounded from above by $m^{\frac{1}{2}+\epsilon}$, i.e., $b_{i,j} = o(m)$.

Lemma 6.5: For any $\epsilon > 0$ the diagonal entries of B are bounded as follows.

$$\mathbb{P}\left(|b_{i,i} - 2(m-1)/3| > 4m^{\frac{1}{2}+\epsilon}\right) \leq 4e^{-m^{2\epsilon}}.$$

The idea is to use Hoeffding's Inequality (see appendix A) for the sum of independent and bounded random variables. To this end, we need to divide the sum in (16) into sums of even and odd terms as follows:

$$b_{i,i} = b_e^i + b_o^i,$$

where

$$b_e^i = 4 \sum_{k \in \text{even}} (x_{k,i} - x_{k+1,i})^2, \quad (18)$$

$$b_o^i = 4 \sum_{k \in \text{odd}} (x_{k,i} - x_{k+1,i})^2. \quad (19)$$

This separation ensures that the random variables in summations (18) and (19) are independent. Let the random variable z_k^i denote the term $4(x_{k,i} - x_{k+1,i})^2$ in (18). Since $z_k^i \in [0, 4]$ and all the terms in b_e^i are independent of each other, we can use Hoeffding's Inequality to upper bound the probability of the deviation of b_e^i from its expected value:

$$\mathbb{P}\left(|b_e^i - (m-1)/3| > 2m^{\frac{1}{2}+\epsilon}\right) \leq 2e^{-m^{2\epsilon}}, \quad (20)$$

for any fixed $\epsilon > 0$. The same bound holds for b_o . Namely,

$$\mathbb{P}\left(|b_o^i - (m-1)/3| > 2m^{\frac{1}{2}+\epsilon}\right) \leq 2e^{-m^{2\epsilon}}. \quad (21)$$

Hence,

$$\begin{aligned} & \mathbb{P}\left(|b_{i,i} - 2(m-1)/3| > 4m^{\frac{1}{2}+\epsilon}\right) \\ & \stackrel{(a)}{\leq} \mathbb{P}\left(|b_e - (m-1)/3| + |b_o - (m-1)/3| > 4m^{\frac{1}{2}+\epsilon}\right) \\ & \stackrel{(b)}{\leq} 4e^{-m^{2\epsilon}}, \end{aligned}$$

where in (a) we used triangular inequality and in (b) we used the union bound.

Lemma 6.6: For any $\epsilon > 0$ the off-diagonal entries of B are bounded as follows.

$$\mathbb{P}\left(|b_{i,j}| > 16m^{\frac{1}{2}+\epsilon}\right) \leq 4e^{-m^{2\epsilon}}.$$

The proof follows in the same lines as the proof of Lemma 6.5.

Using the Gershgorin circle theorem (see appendix A) we can find a lower bound on the minimum eigenvalue of B .

$$\lambda_{\min}(B) \geq \min_i (b_{i,i} - R_i), \quad (22)$$

where

$$R_i = \sum_{j \neq i} |b_{i,j}|.$$

Now, let \mathbb{B}_{ii} denote the event that $\{b_{i,i} < 2(m-1)/3 - 4m^{\frac{1}{2}+\epsilon}\}$ and \mathbb{B}_{ij} (for $i \neq j$) denote the event that $\{b_{i,j} > 16m^{\frac{1}{2}+\epsilon}\}$. Since the matrix B is symmetric, we have only $d(d+1)/2$ degrees of freedom. Lemma 6.5 and 6.6 provide us with a bound on the probability of each event. Therefore, by using the union bound we get

$$\begin{aligned} \mathbb{P}\left(\bigcup_{i \leq j} \overline{\mathbb{B}_{ij}}\right) &\leq 1 - \sum_{i \leq j} \mathbb{P}(\mathbb{B}_{ij}) \\ &= 1 - 3d^2 e^{-m^{2\epsilon}}. \end{aligned}$$

Therefore with probability at least $1 - 3d^2 e^{-m^{2\epsilon}}$ we have

$$b_{i,i} - R_i \geq \frac{2(m-1)}{3} - 16d \cdot m^{\frac{1}{2}+\epsilon}, \quad (23)$$

for all $1 \leq i \leq d$. As m grows, the RHS of (23) can be lower bounded by $(m-1)/3$. By combining (22) and (23) we can conclude that

$$\mathbb{P}\left(\lambda_{\min}(B) \geq \frac{(m-1)}{3}\right) \geq 1 - 3d^2 e^{-m^{2\epsilon}}. \quad (24)$$

As a result, from (26) and (24) we have

$$\mathbb{P}\left(\|(A^T A)^{-1} A\|_2 \leq \sqrt{\frac{3}{m-1}}\right) \geq 1 - 3d^2 e^{-m^{2\epsilon}}, \quad (25)$$

which shows that as m grows, with high probability we have $\|(A^T A)^{-1} A\|_2 \leq \sqrt{\frac{3}{m-1}}$.

F. Proof of Lemmas 6.4 (Deterministic Model)

By using the singular value decomposition of a tall $m-1 \times d$ matrix A , we know that it can be written as $A = U \Sigma V^T$ where U is an orthogonal matrix, V is a unitary matrix and Σ is a diagonal matrix. Then,

$$(A^T A)^{-1} A = U \Sigma^{-1} V^T.$$

Hence,

$$\|(A^T A)^{-1} A\|_2 = \frac{1}{\sigma_{\min}(A)}, \quad (26)$$

where $\sigma_{\min}(A)$ is the smallest singular value of A . This means that in order to upper bound $\|(A^T A)^{-1} A\|_2$ we need to lower bound the smallest singular value of A .

By putting the sensors in the mentioned positions the $d \times d$ matrix A will be Toeplitz and have the following form.

$$A = 2 \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & -1 \\ 0 & \cdots & 0 & 0 & 1 \end{bmatrix}.$$

We can easily find the inverse of matrix A .

$$A^{-1} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 0 & 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 1 \\ 0 & \cdots & 0 & 0 & 1 \end{bmatrix}.$$

Note that the maximum singular value of A^{-1} and the minimum singular value of A are related as follows.

$$\sigma_{\min}(A) = \frac{1}{\sigma_{\max}(A^{-1})}. \quad (27)$$

To find the maximum singular value of A^{-1} , we need to calculate the maximum eigenvalue of $A^{-1} (A^{-1})^T$ which has the following form

$$A^{-1} (A^{-1})^T = \frac{1}{4} \begin{bmatrix} d & d-1 & d-2 & \cdots & 1 \\ d-1 & d-1 & d-2 & \cdots & 1 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 2 & \cdots & 2 & 2 & 1 \\ 1 & \cdots & 1 & 1 & 1 \end{bmatrix}.$$

By using the Gershgorin circle theorem (see appendix A) we can find an upper bound on the maximum eigenvalue of $A^{-1} (A^{-1})^T$.

$$\lambda_{\max} \left(A^{-1} (A^{-1})^T \right) \leq \frac{d^2}{4}, \quad (28)$$

Hence, by combining (26) and (28) we get

$$\|(A^T A)^{-1} A\|_2 \leq \frac{d}{2}. \quad (29)$$

G. Proof of the Bound on the Number of Hops

We start by applying a bin-covering technique in a similar way as in [MP05], [OKM10], [KO10]. In this section, for simplicity, we assume that the nodes are placed in a 3-dimensional space. However, analogous argument proves that the same statement is true for $d = 2$ as well.

For each ordered pair of nodes (i, j) such that $d_{i,j} > R$, define a ‘bin’ as

$$A_{i,j} = \{x \in [0, 1]^3 \mid R - \delta \leq d(x, x_i) \leq R, \angle(x_j - x_i, x - x_i) \leq \theta\},$$

where δ and θ are positive parameters to be specified later in this section, and $\angle(\cdot, \cdot) : \mathbb{R}^d \times \mathbb{R}^d \rightarrow [0, \pi]$ is the angle between two vectors:

$$\angle(\cdot, \cdot) \equiv \arccos(z_1^T z_2 / (\|z_1\| \|z_2\|)).$$

We say a bin $A_{i,j}$ is occupied if there is a node inside the bin that is detected by node i (i.e., connected to node i in the graph G). Next, for each unordered pair of nodes (i, j) such that $d_{i,j} \leq R$, define a bin as

$$B_{i,j} = \{x \in [0, 1]^3 \mid d(x, x_i) \leq R, d(x, x_j) \leq R\}.$$

We say a bin $B_{i,j}$ is occupied if there is a node inside the bin that is simultaneously detected by nodes i and j (i.e., connected to both nodes i and j in the graph G). When n nodes are deployed in $[0, 1]^d$ uniformly at random, we want to ensure that, with high probability, all bins are occupied for appropriate choices of R , δ , and θ .

First when $d_{i,j} > R$,

$$\begin{aligned} \mathbb{P}(A_{i,j} \text{ is occupied}) &= 1 - \prod_{l \neq i,j} (1 - \mathbb{P}(\text{node } l \text{ occupies } A_{i,j})) \\ &\geq 1 - \left(1 - \frac{1}{4} \int_0^\theta \int_{R-\delta}^R 2\pi r^2 \sin(\phi) p(r) dr d\phi\right)^{n-2} \\ &= 1 - \left(1 - \frac{1}{2} \pi \alpha (1 - \cos(\theta)) R^\beta \frac{1}{3-\beta} (R^{3-\beta} - (R-\delta)^{3-\beta})\right)^{n-2}, \end{aligned}$$

for $\beta \in [0, 3)$ as per our assumption. Since $A_{i,j}$ ’s are constrained to be in $[0, 1]^3$, we need to scale the probability by $1/4$. The above inequality is tight in the worst case, that is when both nodes i and j lie on one of the edges of the cube $[0, 1]^3$. We choose θ such that $1 - \cos(\theta) = (\delta/R)^2$. Then using the facts that $1 - z \leq \exp(-z)$ and $(1 - z^{3-\beta}) \leq (3 - \beta)(1 - z)/3$ for $z \in [0, 1)$ and $\beta \in [0, 3)$, we have

$$\mathbb{P}(A_{i,j} \text{ is occupied}) \geq 1 - \exp\left(-\frac{\pi \alpha \delta^3}{6} (n - 2)\right), \quad (30)$$

which is larger than $1 - 1/n^6$ if we set $\delta = (12 \log n / (\alpha(n-2)))^{1/3}$.

Next we consider the case when nodes i and j are at most R apart. Notice that nodes i and j may not be directly connected in the graph G , even if they are within a radio range R . The probability that they are not directly connected is $1 - \alpha(d_{i,j}/R)^{-\beta}$, which does not vanish even for large n . However, we can show that nodes i and j are at most 2 hops apart with overwhelming probability. The event that $h_{i,j} > 2$ is equivalent to the event that $B_{i,j}$ is occupied. Then,

$$\begin{aligned} \mathbb{P}(B_{i,j} \text{ is occupied}) &= 1 - \prod_{l \neq i,j} (1 - \mathbb{P}(\text{node } l \text{ is detected by } i \text{ and } j)) \\ &\geq 1 - (1 - V(B_{i,j})\alpha^2)^{n-2} \\ &\geq 1 - \exp \{ -V(B_{i,j})\alpha^2(n-2) \}, \end{aligned} \quad (31)$$

where $V(B_{i,j}) \in \mathbb{R}$ is the volume of $B_{i,j}$, and we used the fact that the probability of detection is lower bounded by α . $V(B_{i,j})$ is the smallest when nodes i and j are distance R apart and lie on one of the edges of the cube $[0, 1]^3$. In a 3-dimensional space,

$$V(B_{i,j}) \geq (1/4)(5/12)\pi R^3 \geq (1/4)R^3.$$

Substituting these bounds in (31), we get

$$\mathbb{P}(B_{i,j} \text{ is occupied}) \geq 1 - \exp \{ -(1/4)\alpha^2 R^3(n-2) \}, \quad (32)$$

which is larger than $1 - 1/n^6$ for $R \geq ((24 \log n) / ((n-2)\alpha^2))^{1/3}$.

For each ordered pair (i, j) , we are interested in the bin $A_{i,j}$ if $d_{i,j} > R$ and $B_{i,j}$ if $d_{i,j} \leq R$. Using the bounds in (30) and (32) and applying union bound on all $n(n-1)$ ordered pairs of nodes, all bins

$$\{A_{i,j} \mid d_{i,j} > R\} \cup \{B_{i,j} \mid d_{i,j} \leq R\}$$

are occupied with a probability larger than $1 - 1/n^4$.

Now assuming all bins are occupied, we first show that the number of hops between two nodes i and j is bounded by a function $F(d_{i,j})$ that only depends on the distance between the two nodes. The function $F : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ is defined as

$$F(z) = \begin{cases} 2 & \text{if } z \leq R, \\ k+2 & \text{if } z \in \mathcal{L}_k \text{ for } k \in \{1, 2, \dots\}, \end{cases}$$

where \mathcal{L}_k denotes the interval $(k(R - \sqrt{3}\delta) + \sqrt{3}\delta, k(R - \sqrt{3}\delta) + R]$. Our strategy is to use induction to show that for all pairs of nodes,

$$h_{i,j} \leq F(d_{i,j}). \quad (33)$$

First, assume nodes i and j are at most R apart. Then, by the assumption that the bin $B_{i,j}$ is occupied there is a node connected to both i and j . Therefore the number of hops $h_{i,j}$ is at most 2.

Next, assume that the bound in (33) is true for all pairs (l, m) with

$$d_{l,m} \leq \sqrt{3}\delta + k(R - \sqrt{3}\delta).$$

For two nodes i and j at distance $d_{i,j} \in \mathcal{L}_k$, consider a line segment $\ell_{i,j}$ in the 3-dimensional space with one end at x_i and the other at x_j . Let $y \in \mathbb{R}^3$ be the point in the line segment $\ell_{i,j}$ that is at distance R from x_i . We want to show that there exists a node that is close to y and is connected to node i . By definition, y is inside the bin $A_{i,j}$. We know that the bin $A_{i,j}$ is occupied by at least one node that is connected to node i . Let us denote one of these nodes by l . Then $d(y, x_l) \leq \sqrt{3}\delta$ because

$$\sup_{z \in A_{i,j}} d(z, y) = \sqrt{\delta^2 + 2R(R - \delta)(1 - \cos(\theta))} \leq \sqrt{3}\delta.$$

We use the following triangular inequality which follows from the definition of the number of hops.

$$h_{i,j} \leq h_{i,l} + h_{l,j}.$$

Since l is connected to i we have $h_{i,l} = 1$. By triangular inequality, we also have $d_{l,j} \leq d(y, x_j) + d(y, x_l)$.

It follows from $d(y, x_j) = d_{i,j} - R$ and $d(y, x_l) \leq \sqrt{3}\delta$ that

$$d_{l,j} \leq d_{i,j} - R + \sqrt{3}\delta.$$

Recall that we assumed $d_{i,j} \leq R + k(R - \sqrt{3}\delta)$. Since we assumed that (33) holds for $d_{l,j} \leq \sqrt{3}\delta + k(R - \sqrt{3}\delta)$, we have

$$h_{i,j} \leq k + 2,$$

for all nodes i and j such that $d_{i,j} \leq R + k(R - \sqrt{3}\delta)$. By induction, this proves that the bound in (33) holds for all pairs (i, j) .

We can upper bound $F(z)$ with a simple affine function:

$$\begin{aligned} F(z) &\leq 2 + \frac{1}{R - \sqrt{3}\delta} z \\ &\leq 2 + \left(1 + \frac{2\delta}{R}\right) \frac{z}{R}, \end{aligned}$$

where the last inequality is true for $R \geq 2\sqrt{3}\delta/(2 - \sqrt{3})$. Together with (33) this finishes the proof of the lemma.

Figure 13 illustrates the comparison of the upper bounds $F(d_{i,j})$ and $F_a(d_{i,j})$, and the trivial lower bound $\hat{d}_{i,j} \geq d_{i,j}$ in a simulation with parameters $d = 2$, $n = 6000$ and $R = \sqrt{64 \log n / n}$. The

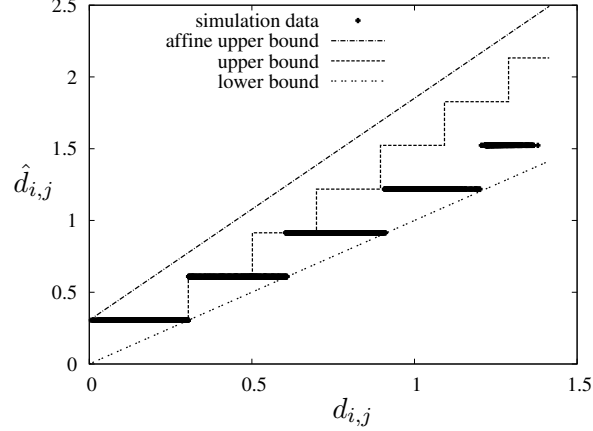


Fig. 13. Comparison of upper and lower bound of shortest paths $\{\hat{d}_{i,j}\}$ with respect to the correct distance $\{d_{i,j}\}$ computed for $n = 6000$ sensors in 2-dimensional square $[0, 1]^2$ under connectivity-based model.

simulation data shows the distribution of shortest paths between all pairs of nodes with respect to the actual pairwise distances, which confirms that the shortest paths lie between the analytical upper and lower bounds. Although the gap between the upper and lower bound is seemingly large, in the regime where $R = C\sqrt{\log n/n}$ with a constant C , the vertical gap R vanishes as n goes to infinity and the slope of the affine upper bound can be made arbitrarily small by increasing the radio range R or equivalently taking large enough C .

VII. CONCLUSION

In many applications of wireless sensor networks, it is crucial to determine the location of nodes. For this matter, numerous algorithms have been recently proposed where the efficiency and success of them have been mostly demonstrated by simulations. In this paper, we have investigated the centralized and distributed sensor localization problem from a theoretical point of view and have provided analytical bounds on the performance of such algorithms. More precisely, we analysed the MDS-MAP and HOP-TERRAIN algorithms and showed that even when only the connectivity information was given and in the presence of detection failure, the resulting error of both algorithms is bounded and decays at a rate inversely proportional to the detection range.

ACKNOWLEDGMENT

We would like to thank Andrea Montanari and Rüdiger Urbanke for their stimulating discussions on the subject of this paper.

REFERENCES

- [BG05] I. Borg and P.J.F. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer, 2005.
- [BY04] P. Biswas and Y. Ye. Semidefinite programming for ad hoc wireless sensor network localization. In *IPSN '04: Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 46–54, New York, NY, USA, 2004. ACM.
- [CC01] T. F. Cox and M. A. A. Cox. *Multidimensional Scaling*. Chapman & Hall, 2001.
- [CHH02] Srdjan Capkun, Maher Hamdi, and Jean-Pierre Hubaux. GPS-free Positioning in Mobile Ad-Hoc Networks. 2002.
- [CR08] Emmanuel J Candès and Benjamin Recht. Exact Matrix Completion via Convex Optimization. 2008.
- [DJMI⁺06] P. Drineas, A. Javed, M. Magdon-Ismail, G. Pandurangant, R. Virrankoski, and A. Savvides. Distance matrix reconstruction from incomplete distance information for sensor network localization. In *Proceedings of Sensor and Ad-Hoc Communications and Networks Conference (SECON)*, volume 2, pages 536–544, Sept. 2006.
- [Faz02] M Fazel. Matrix rank minimization with applications. *Elec Eng Dept Stanford University*, 2002.
- [GHDT10] Steven J. Gortler, Alexander D. Healy, Dylan, and P. Thurston. Characterizing generic global rigidity, 2010.
- [GK98] P. Gupta and P.R. Kumar. Critical power for asymptotic connectivity. In *Proceedings of the 37th IEEE Conference on Decision and Control*, volume 1, pages 1106–1110 vol.1, 1998.
- [HJ85] R. Horn and C. Johnson. *Matrix analysis*. Cambridge University Press, 1985.
- [Hoe63] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:1330, 1963.
- [IFMW04] A. T. Ihler, J. W. Fisher, R. L. Moses, and A. S. Willsky. Nonparametric belief propagation for self-calibration in sensor networks. In *IPSN '04: Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 225–233, New York, NY, USA, 2004. ACM.
- [JH01] Gaetano Borriello Jerrey Hightower. Location Systems for Ubiquitous Computing. 2001.
- [JJ05] Bill Jackson and Tibor Jordán. Connected rigidity matroids and unique realizations of graphs. *J. Comb. Theory Ser. B*, 94:1–29, May 2005.
- [JM11] Adel Javanmard and Andrea Montanari. Localization from Incomplete Noisy Distance Measurements. *arXiv.org*, 2011.
- [Joh77] D. B. Johnson. Efficient algorithms for shortest paths in sparse networks. *J. ACM*, 24(1):1–13, 1977.
- [KM06] AA Kannan and G Mao. Simulated annealing based wireless sensor network localization. *Journal of Computers*, 2006.
- [KM10] RH Keshavan and A Montanari. Matrix completion from noisy entries. *The Journal of Machine Learning*, 2010.
- [KMV06] A.A Kannan, Guoqiang Mao, and B Vucetic. Simulated Annealing based Wireless Sensor Network Localization with Flip Ambiguity Mitigation. In *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, pages 1022–1026, 2006.
- [KO10] A. Karbasi and S. Oh. Distributed sensor network localization from local connectivity: Performance analysis for the HOP-TERRAIN algorithm. In *ACM SIGMETRICS*, June 2010.
- [KOPV10] Amin Karbasi, Sewoong Oh, Reza Parhizkar, and Martin Vetterli. Ultrasound Tomography Calibration Using Structured Matrix Completion. In *The 20th International Congress on Acoustics*, 2010.
- [Led01] Michel Ledoux. *The Concentration of Measure Phenomenon*. American Mathematical Society, 2001.
- [LR03] K. Langendoen and N. Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. *Comput. Netw.*, 43(4):499–518, 2003.

- [MFA07] G MAO, B FIDAN, and B ANDERSON. Wireless sensor network localization techniques. *Computer Networks*, 51(10):2529–2553, July 2007.
- [MP05] S. Muthukrishnan and G. Pandurangan. The bin-covering technique for thresholding random geometric graph properties. In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 989–998, Philadelphia, PA, USA, 2005.
- [MWY06] AS Morse, W Whiteley, and YR Yang. A Theory of Network Localization. *IEEE TRANSACTIONS ON MOBILE COMPUTING*, 2006.
- [NN01] D. Niculescu and B. Nath. Ad hoc positioning system (aps). In *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, pages 2926–2931, 2001.
- [NN03] D. Niculescu and B. Nath. DV based positioning in ad hoc networks. *Journal of Telecommunication Systems*, 22:267–280, 2003.
- [NSB03] R. Nagpal, H. Shrobe, and J. Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. In *IPSN '03: Proceedings of the 2nd international conference on Information processing in sensor networks*, pages 333–348, 2003.
- [OKM10] S. Oh, A. Karbasi, and A. Montanari. Sensor network localization from local connectivity : performance analysis for the MDS-MAP algorithm. In *Proc. of the IEEE Inform. Theory Workshop*, January 2010.
- [PCB00] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 32–43, New York, NY, USA, 2000. ACM.
- [RS00] S T Roweis and L K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science (New York, N.Y.)*, 290(5500):2323–2326, December 2000.
- [SHS01] A. Savvides, C. Han, and M. B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 166–179, New York, NY, USA, 2001. ACM.
- [SLR02] C. Savarese, K. Langendoen, and J. Rabaey. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. In *USENIX Technical Annual Conference*, pages 317–328, Monterey, CA, June 2002.
- [SPS03] A Savvides, H Park, and M. Srivastava. The n-hop multilateration primitive for node localization problems. *Mob. Netw. Appl.*, 8(4):443–451, 2003.
- [SR03] Lawrence K. Saul and Sam T. Roweis. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *The Journal of Machine Learning Research*, 2003.
- [SRB01] C. Savarese, J. Rabaey, and J. Beutel. Locationing in distributed ad-hoc wireless sensor networks. In *in ICASSP*, pages 2037–2040, 2001.
- [SRZF03] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz. Localization from mere connectivity. In *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 201–212, New York, NY, USA, 2003. ACM.
- [SRZF04] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz. Localization from connectivity in sensor networks. *IEEE Trans. Parallel Distrib. Syst.*, 15(11):961–974, 2004.
- [Xu02] Ning Xu. A survey of sensor network applications. *IEEE Communications Magazine*, 40, 2002.

APPENDIX

Hoeffding's inequality [Hoe63] is a result in probability theory that gives an upper bound on the probability for the sum of random variables to deviate from its expected value. Let z_1, z_2, \dots, z_n be independent and bounded random variables such that $z_k \in [a_k, b_k]$ with probability one. Let $s_n = \sum_{k=1}^n z_k$. Then for any $\delta > 0$, we have

$$\mathbb{P}(|s_n - \mathbb{E}[s_n]| \geq \delta) \leq 2 \exp\left(-\frac{2\delta^2}{\sum_{k=1}^n (b_k - a_k)^2}\right).$$

The **Gershgorin circle theorem** [HJ85] identifies a region in the complex plane that contains all the eigenvalues of a complex square matrix. For an $n \times n$ matrix A , define

$$R_i = \sum_{j \neq i} |a_{i,j}|.$$

Then each eigenvalue of A is in at least one of the disks

$$\{z : |z - a_{i,i}| \leq R_i\}.$$

Informally, **concentration of Lipschitz functions** says that any *smooth* function of bounded independent random variables is tightly concentrated around its expectation [Led01]. The notion of smoothness we will use is *Lipschitz*.

Definition A.1: $f : \mathbb{R}^n \rightarrow R$ is λ -Lipschitz with respect to the l_p norm, if for all x and y ,

$$|f(x) - f(y)| \leq \lambda \|x - y\|_p.$$

It turns out that Hoeffding's bound holds for all Lipschitz (with respect to l_1 norm) functions. More precisely, suppose X_1, X_2, \dots, X_n are independent and bounded with $a_i \leq x_i \leq b_i$. Then for any $f : \mathbb{R}^n \rightarrow R$ which is λ -Lipschitz with respect to the l_1 norm,

$$\Pr(|f - E(f)| \geq \epsilon) \leq 2 \exp\left(-\frac{2\epsilon^2}{\lambda^2 \sum_{i=1}^n (b_i - a_i)^2}\right).$$